Common Security Protocol Structure and Mechanism and System and Method for Using Common Security Protocol

(A-70556/RMA)

WE CLAIM:

1. A computer program product for use in conjunction with a computer system having a server and a client, the computer program product comprising a computer readable storage medium and a computer program mechanism embedded therein, the computer program mechanism, comprising: a program module that directs the computer system and/or components thereof including at least one or the client or server, to function in a specified manner to provide message communications, the message communications occurring in a computer system hardware architecture neutral and operating system neutral and network transport protocol neutral manner for implementing a plurality of separate security protocols using a common set of criteria, the program module including instructions for:

A. defining two cryptographic primitives; and

B. using only said two cryptographic primitives to construct said plurality of separate security protocols.

2. A hardware architecture neutral and operating system neutral and network transport neutral method for implementing a plurality of separate security protocols using a common set of criteria, said method comprising the steps of:

A. defining two cryptographic primitives; and

B. using only said two cryptographic primitives to construct said plurality of separate security protocols.

3. The method in claim 2, wherein said two cryptographic primitives are sued to construct a greater plurality of security protocols.

4. The method in claim 2, wherein said cryptographic primitives including formats and algorithms.

5. The method in claim 2, wherein said cryptographic primitives consist of only formats and algorithms.

6. The method in claim 2, wherein said cryptographic primitives being for: (i) Encrypted-Data, and for (ii) Signed-Inside-Enveloped-Data.

7. The method in claim 6, wherein said cryptographic primitives for Encrypted-Data providing privacy and data integrity based on a secret key and a cipher algorithm.

8. The method in claim 7, wherein said cipher algorithm being selected from the group of cipher algorithms consisting of triple-DES, XTEA, RC4, AES, block cipher algorithms, stream ciphers, and combinations thereof.

9. The method in claim 6, wherein said cryptographic primitives for Signed-Inside-Enveloped-Data providing transport of a secret key from Sender to Recipient using a public key of the recipient.

10. The method in claim 9, wherein said secret key being selected from the set comprising a message key and a session key.

11. The method in claim 9, wherein said signed-inside-enveloped-data further providing data privacy plus integrity using the Encrypted-Data primitive and providing data authenticity using a public key digital signature and provides the certificate chain of the Sender.

12. The method in claim 6, wherein said cryptographic primitives for Encrypted-Data providing privacy and data integrity based on a secret key and a cipher algorithm; and said cryptographic primitives for Signed-Inside-Enveloped-Data providing transport of a secret key from Sender to Recipient using a public key of the recipient.

13. The method in claim 2, wherein said security protocols are selected from the group consisting of: (i) secure interactive sessions, (ii) secure unidirectional messaging, (iii) secure software downloading, (iv) secure software upgrading, (v) secure issuing of digital certificates, and/or (vi) combinations thereof.

14. The method in claim 2, wherein the common set of criteria are selected from the set consisting of data formats, algorithms, subroutines, procedures, and combinations thereof.

15. The method in claim 6, wherein said cryptographic primitives for Encrypted-Data providing privacy and data integrity based on a secret key and a cipher algorithm.

16. The method in claim 7, wherein said cipher comprise a block cipher; the primitive includes an Initialization Vector for Cipher-Block-Chaining mode that is an input to the primitive and appears in the data format of the output; and, the primitive returns a new Initialization Vector to be used with the next block of Encrypted Data.

17. The method in claim 16, wherein the secret key to the cipher is one input to this primitive.

18. The method in claim 16, wherein said block cipher is a cipher selected from the set consisting of a triple-DES based cipher, and a XTEA based cipher.

19. The method in claim 7, wherein said cipher comprise a stream cipher without an Initialization Vector, the bytes of the key are not reused, and the secret key to the cipher is one input to this primitive.

20. The method in claim 19, wherein said stream cipher comprises a RC4 type cipher.

21. The method in claim 2, wherein the integrity of the data and associated data tamper detection, is provided by a cryptographic message authentication code that is based on a secret key.

22. The method in claim 21, wherein the secret is equal to or derived from the key used to encrypt the data.

23. The method in claim 22, the authentication code is computed by a CBC-MAC based algorithm and/or a HMAC based algorithm.

24. The method in claim 2, wherein the primitive takes as an optional input some other data that is protected by the cryptographic message authentication code, but not part of the output data.

25. The method in claim 24, wherein such other data is selected from the set of data identified as data in a Type Field, Version Field, Content-Length field, and combinations thereof.

26. The method in claim 25, wherein said cryptographic primitives include primitives for Encrypted-Data and for Signed-Inside-Enveloped-Data; and the Type field is transmitted first before the Encrypted-Data and not be part of the Encrypted-Data.

27. The method in claim 2, wherein said using only said two primitives to construct a plurality of separate security protocols further comprises using fixed public keys and/or certificates when a protocol application does not have, does not use, or does not require public keys and/or certificates for both the Sender and the Recipient.

28. The method in claim 27, wherein for a protocol application that does not require that the data be encrypted, using Signed-Inside-Enveloped-Data to provide the software signing, and using a fixed Recipient public key to which all receiving software knows the private key for the encryption, rather than providing a special third cryptographic primitive for signed-only data as is done in some conventional systems is such circumstances.

29. The method in claim 28, wherein said protocol application includes downloading signed software.

30. The method in claim 2, wherein said using only said two primitives to construct a plurality of separate security protocols further comprise including both signing and encryption public keys in the certificates used with this protocol so it is possible to send an encrypted message back to the Sender of a message.

31. The method in claim 2, wherein the Signed-Inside-Enveloped-Data primitive provides all the security functions required for secure unidirectional messaging.

32. The method in claim 31, wherein said unidirectional messaging includes electronic mail (e-mail).

33. The method in claim 6, wherein the Signed-Inside-Enveloped-Data primitive provides a component for setting up a session key with a new entity for which the Sender knows the Recipient's public key.

34. The method in claim 33, wherein the Sender knows the recipient's public key by any one of: (i) a plain text request of the certificate of the Recipient, (ii) by sending the Recipient a master secret from which the session keys are derived, or (iii) by the Sender having received the Recipient's certificate in a previous communication.

35. The method in claim 6, wherein the keys for the Encrypted-Data primitive are derived from exchanged information.

36. The method in claim 35, wherein the exchanged information is information exchanged either in the clear, or information exchanged in the Signed-Inside-Enveloped-Data primitive.

37. The method in claim 36, wherein said information exchanged in the clear comprises non-secure plain text.

38. The method in claim 35, wherein said keys for the Encrypted-Data primitive derived from exchanged information provides a form of dual key determination and challenge-response authentication.

39. The method in claim 6, wherein new secret session keys are derived from old secret keys that where previously agreed to by the Sender and Recipient thereby avoiding all or a component of overhead of public and private key operations by just using the Encrypted-Data primitive with the appropriate keys.

40. The method in claim 6, wherein authentication for a session key is provided by using the Encrypted-Data primitive with values that are produced by the cryptographic hash of some or all of the data transmitted before sending the authentication message.

41. The method in claim 40, wherein all of the prior data transmitted is included to help thwart attacks on cryptographic protocols.

42. The method in claim 6, wherein, to avoid various protocol attacks, separate keys are used by the Sender and Recipient by deriving the keys in different ways from shared information exchanged earlier in the protocol and/or fixed information known to the Sender and Recipient.

43. The method in claim 13, wherein certificate issuing is authenticated by sending a Resource Tag to the Issuer after the session keys have been established.

44. The method in claim 43, wherein the fixed public and private keys are replaced with the newly generated keys once the client has received the Certificate keys.

45. The method in claim 44, wherein said newly generated keys being generated either on the client or by the Issuer.

46. The method in claim 43, wherein the fixed public and private keys are replaced with the newly generated keys once the client has received the Certificate and the keys.

47. The method in claim 43, wherein said Resource Tag comprises a Message Tag or a Coupon Tag.

48. The method in claim 13, wherein the certificate issuing is further authenticated using fixed public and private keys for the client device that wants to get a Certificate from the Issuer.

49. The method in claim 6, wherein a Secure Response message protocol is implemented using the Signed-Inside-Enveloped-Data primitive with a public key of the Recipient that is included inside the message to which this is a response.

50. The method in claim 49, wherein said message is a promotional message.

51. The method in claim 49, wherein the message includes a Certificate and the Signed-Inside-Enveloped-Data primitive with a public key of the Recipient is inside the Certificate that is verified by the Sender of the Response.

52. The method in claim 49, wherein this Secure Response message protocol is either a unidirectional response message or the set up portion of a bi-directional messaging session.

53. The method in claim 49, wherein the Secure Response message protocol is implemented using the Encrypted-Data primitive with a secret key know to the Recipient that is included inside the message that was received securely.

54. The method in claim 49, wherein the Secure Response message protocol is implemented using the Encrypted-Data primitive with a secret key know to the Recipient that is included inside the message that was received securely and the Encrypted-Data primitive containing the Response Message.

55. The method in claim 53, wherein this Secure Response message protocol is either a unidirectional response message or the set up portion of a bi-directional session.

56. The method in claim 54, wherein this Secure Response message protocol is either a unidirectional response message or the set up portion of a bi-directional session.